

**Joint Tactical Radio System (JTRS) Standard
Device Message Control
Application Program Interface (API)**



**Version: 1.1.3
31 March 2008**

Statement A- Approved for public release; distribution is unlimited (29 March 2007)

REVISION HISTORY

Version	Authorization	Description	Last Modified Date
1.0		Initial release ICWG Approved	23-January-2006
1.1		Update outline format ICWG Approved	26-January-2006
1.1.1		Preparation for public release	29-March-2007
1.1.2 <Draft>		Change Proposal: -Specified “purge” and “endOf Stream” applicability	20-August-2007
1.1.2		Updates based on ICWG review ICWG Approved	01-October-2007
1.1.3		Public Release	31-March-2008

Table of Contents

A. DEVICE MESSAGE CONTROL.....	6
--------------------------------	---

Table of Contents

A. DEVICE MESSAGE CONTROL.....	6
A.1 Introduction.....	6
A.1.1 Overview.....	6
A.1.2 Service Layer Description.....	7
A.1.3 Modes of Service.....	7
A.1.4 Service States.....	7
A.1.5 Referenced Documents.....	8
A.1.5.1 Government Documents.....	8
A.1.5.2 Commercial Standards.....	8
A.2 Services.....	9
A.2.1 Provide Services.....	9
A.2.2 Uses Services	9
A.2.3 Interface Modules.....	9
A.2.3.1 DevMsgCtl	9
A.2.4 Sequence Diagrams	10
A.2.4.1 Abort Transmit Data Sequence.....	10
A.3 Service Primitives and Attributes	11
A.3.1 DevMsgCtl::DeviceMessageControl.....	12
A.3.1.1 <i>abortTx</i> Operation.....	12
A.3.1.2 <i>rxActive</i> Operation	13
A.3.1.3 <i>txActive</i> Operation	14
A.4 IDL.....	15
A.4.1 DeviceMessageControl IDL	15
A.5 UML.....	16
Appendix A.A Abbreviations and Acronyms.....	17
Appendix A.B Performance Specification	18

Lists of Figures

FIGURE 1 – DEVICEMESSAGECONTROL INTERFACE DIAGRAM	9
FIGURE 2 – ABORT TRANSMIT DATA SEQUENCE DIAGRAM	10

A. DEVICE MESSAGE CONTROL

A.1 INTRODUCTION

This document defines a common set of *Device Message Control* interfaces to be used by Joint Tactical Radio (JTR) Set Applications and Services. The *Device Message Control* interface provides control for communications between a receiver and transmitter. It allows the *Device User* to indicate when the transmission is active and also provides an interface for aborting the transmission.

The *Device Message Control* interfaces are documented within to minimize coupling between the device and service interfaces that utilize these *Device Message Control* interfaces.

A.1.1 Overview

This document contains as follows:

- a. Section A.1, *Introduction*, contains the introductory material regarding the Overview and Referenced Documents of this document.
- b. Section A.2, *Services*, specifies the interface, port connections, and sequence diagrams.
- c. Section A.3, *Service Primitives and Attributes*, specifies the operations that are provided by the *Device Message Control* interface.
- d. Section A.4, *IDL*.
- e. Section A.5, *UML*.
- f. Appendix A.A, *Abbreviations and Acronyms*.
- g. Appendix A.B, *Performance Specification*.

A.1.2 Service Layer Description

Not applicable.

A.1.3 Modes of Service

Not applicable.

A.1.4 Service States

Not applicable.

A.1.5 Referenced Documents

The following documents of the exact issue shown form a part of this specification to the extent specified herein.

A.1.5.1 Government Documents

A.1.5.1.1 Specifications

A.1.5.1.1.1 Federal Specifications

None

A.1.5.1.1.2 Military Specifications

None

A.1.5.1.2 Other Government Agency Documents

None

A.1.5.2 Commercial Standards

None

A.2 SERVICES

A.2.1 Provide Services

Not applicable.

A.2.2 Uses Services

Not applicable.

A.2.3 Interface Modules

A.2.3.1 DevMsgCtl

A.2.3.1.1 DeviceMessageControl Interface Description

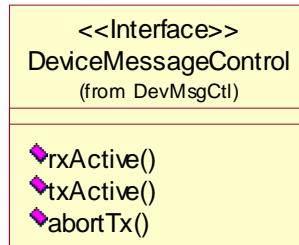


Figure 1 – DeviceMessageControl Interface Diagram

The interface design of *DeviceMessageControl* is shown in Figure 1. It allows the *Device User* to indicate when the transmission is active and also provides an interface for aborting the transmission.

A.2.4 Sequence Diagrams

A.2.4.1 Abort Transmit Data Sequence

Description

A *Device User* aborts the transmit data stream as shown in Figure 2. After the *abortTx* is called, all data is purged from the device. The *Device* will return a 0 length *pushPacket* with the “purge” and “endOfStream” flags set to TRUE.

Note: The “purge” and “endOfStream” response behavior for the *abortTx ()* is only applicable for streaming devices that require it.

Pre-conditions

The *Device* is in the ENABLED state.

Post-conditions

The transmit data is aborted.

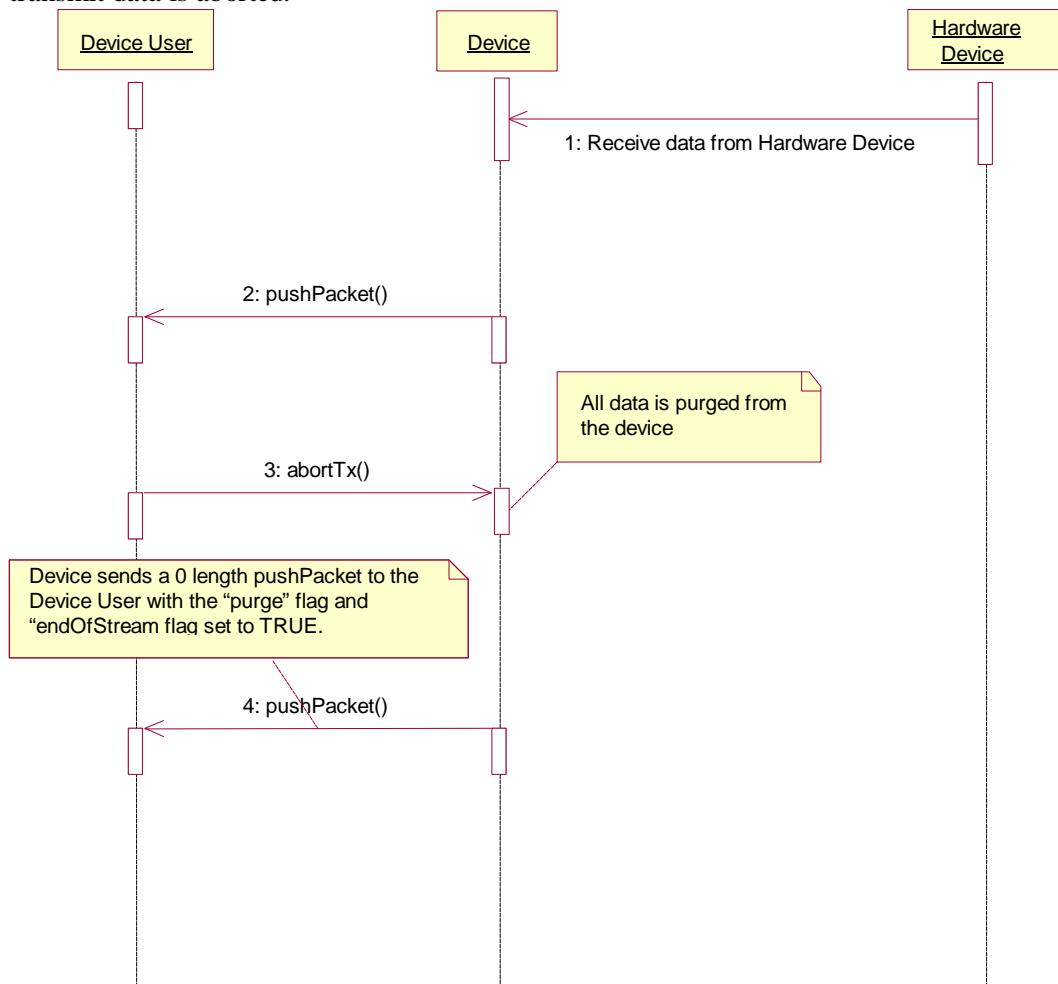


Figure 2 – Abort Transmit Data Sequence Diagram

A.3 SERVICE PRIMITIVES AND ATTRIBUTES

To enhance the readability of this API document and to avoid duplication of data, the type definitions of all structured types (i.e., data types, enumerations, exceptions, and structures) used by the Service Primitives and Attributes have been co-located in section A.5. This cross-reference of types also includes any nested structures in the event of a structure of structures or an array of structures.

A.3.1 DevMsgCtl::DeviceMessageControl

A.3.1.1 *abortTx* Operation

The *abortTx* operation aborts the specified stream if in progress. If aborted, processing will terminate and the stream will be purged. See section A.2.4.1.

Note: The “purge” and “endOfStream” response behavior for the *abortTx()* is only applicable for streaming devices that require it.

A.3.1.1.1 Synopsis

void abortTx(in unsigned short streamId);

A.3.1.1.2 Parameters

Parameter Name	Description	Type
streamId	The identification number of the stream to be aborted.	unsigned short

A.3.1.1.3 State

Not applicable.

A.3.1.1.4 New State

Not applicable.

A.3.1.1.5 Return Value

None

A.3.1.1.6 Originator

Not applicable.

A.3.1.1.7 Exceptions

None

A.3.1.2 *rxActive* Operation

The *rxActive* operation indicates if the supporting component is actively processing Rx traffic. Rx processing begins with the receipt of the first packet in a stream, and ends when the last packet of data has been passed through the device (empty condition).

A.3.1.2.1 Synopsis

boolean rxActive();

A.3.1.2.2 Parameters

None

A.3.1.2.3 State

Not applicable.

A.3.1.2.4 New State

Not applicable.

A.3.1.2.5 Return Value

Description	Type
The status of whether the device is transferring data in the receive direction.	boolean

A.3.1.2.6 Originator

Not applicable.

A.3.1.2.7 Exceptions

None

A.3.1.3 *txActive* Operation

The *txActive* operation indicates if the supporting component is actively processing Tx traffic. Tx processing begins when the component has begun processing (typically before the first packet is sent downstream) and ends when the last packet of data has been passed to the consumer.

A.3.1.3.1 Synopsis

boolean txActive();

A.3.1.3.2 Parameters

None

A.3.1.3.3 State

Not applicable.

A.3.1.3.4 New State

Not applicable.

A.3.1.3.5 Return Value

Description	Type
The status of whether the device is transferring data in the transmit direction.	boolean

A.3.1.3.6 Originator

Not applicable.

A.3.1.3.7 Exceptions

None

A.4 IDL

A.4.1 DeviceMessageControl IDL

```
/*
** DeviceMessageControl.idl
*/

#ifndef __DEVICEMESSAGECONTROL_DEFINED
#define __DEVICEMESSAGECONTROL_DEFINED

module DevMsgCtl {

    interface DeviceMessageControl {
        boolean rxActive ();
        boolean txActive ();
        void abortTx (in unsigned short streamId);
    };
}

#endif
```

A.5 UML

Not applicable.

APPENDIX A.A ABBREVIATIONS AND ACRONYMS

API	Application Program Interface
ICWG	Interface Control Working Group
IDL	Interface Definition Language
JPEO	Joint Program Executive Office
JTR	Joint Tactical Radio
JTRS	Joint Tactical Radio System
Rx	Receive
Tx	Transmit
UML	Unified Modeling Language

APPENDIX A.B PERFORMANCE SPECIFICATION

Not applicable.